WHAT IS CLAIMED IS:

1. A method for interactive volume rendering of substantial amounts of volume data in form of a stack of original 2-dimensional slices into displayable images on a display of a personal computer, said personal computer having at least one graphics processing unit, comprising:

reconstructing a 3-dimensional texture map of said volume data from processed 2-dimensional slices taken from said original 2-dimensional slices;

segmenting said 3-dimensional texture map into three stacks of 3-dimensional-textured slices;

rescaling said 3-dimensional-textured slices so that each slice edge dimension is of an integer power of two to yield rescaled 2-dimensional slices;

subdividing each said rescaled 2-dimensional slice into grids of blocks with corresponding depth and texture coordinate information;

in response to input designating view and size of image display of said volume data, carrying out selected transformations, including at least translation, rotation, scaling and plane-clipping, on said grids of blocks;

performing a two-pass rendering process on said grids of blocks comprising a virtual rendering pass in order to compute information of view-dependent unused blocks, and a main rendering pass in order to obtain processed blocks for further filtration; and

applying block-based fragment filtration to the processed blocks to obtain image elements suited for display and to render a final image.

2. The method according to claim 1 wherein said 3-dimensional-textured slices are axis-aligned.

3. The method according to claim 1 wherein the slice subdividing step comprises:

dividing each said rescaled 2-dimensional slice into a grid of regular square blocks of smaller texture, the edge dimension of each said block being of an integer power of two., while associating an index with each said block.

4. The method according to claim 3 further including the step of storing vertex coordinates and corresponding texture coordinates of said blocks.

5. The method according to claim 1 wherein

said virtual rendering pass includes rendering said volume data to compute view dependent visibility information, and storing said visibility information in system memory; and wherein

said main rendering pass includes static block filtration and dynamic block filtration while rendering said final image;

storing current rendering status, including at least current translation status, current rotation status, current scaling status and current plane-clipping status in the system memory; and

sharing said current rendering status between said main rendering pass and said virtual rendering pass.

6. The method according to claim 5 wherein

a main rendering thread is allocated to a single main graphics slot, and at least one virtual rendering thread is allocated to side graphics slots.

7. The method according to claim 6 wherein said main rendering thread and at least one said virtual rendering thread are distributed among a plurality of graphics processing units.

8. The method according to claim 5 wherein said virtual volume rendering step includes:

selecting a corresponding stack out of three said axis-aligned grids of blocks according to current translational status and current rotational status of said volume data;

retrieving vertex information of every said block;

storing identity of every block within said corresponding stack as color texture;

applying any clipping planes onto the rendering procedure;

combining color texture, alpha texture and vertex buffer to yield combined texture; and

rendering said combined texture to a virtual rendered item buffer, in order to compute information of any non-viewable blocks in preparation for transferring identity of viewable blocks of the virtual screen buffer to the system memory.

9. The method according to claim 8 wherein the combined texture rendering step is a multi-GPU process using a plurality of vertex shaders and fragment shaders in said virtual rendering.

10. The method according to claim 9 wherein the multi-GPU process includes:

dividing said grids of blocks into different sets according an available number of side graphics slots;

rendering different sets of slices to the virtual screen individually;

merging all visibility information; and

copying the resultant merged visibility information to system memory.

11. The method according to claim 5 wherein said main rendering pass comprises:

performing the static block filtration to filter out view-independent blocks including merely non-contributing signals and to obtain statically filtrated blocks; and

performing the dynamic block filtration to filter out view dependent blocks due to occlusion.

12. The method according to claim 11 wherein, during the static block filtration, each block in three said axis-aligned grids of blocks is processed to reduce unused data, including:

providing the non-contributing signals as a set of specific color entries as a filter set;

performing the static block filtration on each block in said grid of blocks to filter out the non-contributing signals from the rasterization process according to said filter set;

identifying blocks as to-be-removed if and only if the whole block is filled only with colors from said filter set; and

recording indices of statically filtrated blocks.

13. The method according to claim 11 wherein the dynamic block filtration step includes:

selecting a corresponding stack out of the three said axis-aligned grids of blocks according to current translational status and current rotational status of said volume data;

retrieving vertex information of said statically filtrated blocks;

reading the visibility information and a current projection matrix from the system memory;

determining filtrate-blocks using the visibility information;

directing a vertex buffer of said filtrate-blocks to the vertex processor for rasterization and processed textures coordinates to the fragment processor; and

rendering the final image by said vertex buffer with said 3-dimensional texture map.